Comparitive study to classify presence of heart disease in individual using KNN and Decision Tree Classifier.

Ameya Naik

Abstract

Summarize your questions and findings in 1 brief paragraph (4-6 sentences max). Your abstract needs to include: what dataset, what question, what method was used, and findings.

I have referred to Heart Disease dataset available at the below link :

https://archive.ics.uci.edu/ml/datasets/heart+disease

In this project , we would classify presence of heart disease on the basis of given features.

We have also splitted the data set in train-test to check accuracy of two Classification algorithm that we are apply(KNN and Decision Tree Classifier)

We will also determine how algorithms stand in comparison to each other , and which is the best to accurates classify.

Motivation

Describe the problem you want to solve with the data. It may relate closely with your research question, but your goal here is to make your audience care about the project/problem you are trying to solve. You need to articulate the problem you are exploring and why (and for whom) insight would be valuable.

We would be classifying the presence of heart disease on the basis of the 13 most important scientifically proven important indicators.

This would help to detect prescen for future patient , to determine onset of heart disease , hence preventive measure could be taken , to avoid worsening the condition.

Dataset(s)

Describe your dataset(s) here. You should say what data is in the dataset, how much data, and where you found the dataset (if applicable).

The data set isn’t huge hence perfect to apply KNN. It contains most relevant feature to detemoine heart disease like :

By looking on the head , we can see which features are categorical and which are numerical.

Categorical : sex, cp , fbs restecg, exang slope

sex: sex (1 = male; 0 = female)

cp: chest pain type

-- Value 1: typical angina

-- Value 2: atypical angina

-- Value 3: non-anginal pain

-- Value 4: asymptomatic

trestbps: resting blood pressure (in mm Hg on admission to the hospital)

chol: serum cholestoral in mg/dl

fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)

ca: number of major vessels (0-3) colored by flourosopy

thalach: maximum heart rate achieved

exang: exercise induced angina (1 = yes; 0 = no)

oldpeak : ST depression induced by exercise relative to rest

slope: the slope of the peak exercise ST segment

-- Value 1: upsloping

-- Value 2: flat

-- Value 3: downsloping

thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

Data Preparation and Cleaning

At a high-level, what did you need to do to prepare the data for analysis? Describe what problems, if any, did you encounter with the dataset?

The data had lots of features which weren’t relevant , hence I did drop all the irrelevant ones and kept the 14 features which were important for analysis.

Luckily , there weren’t any null values in the data hence , I didn’t need to drop or impute any row.

Research Question(s)

What is your research question you aim to answer using the dataset? Be sure the research question is well defined (see project description for details).

Is there a correlation between thalach (maximum heart rate achivied) , to heart disease and if there is how strong?

What gender individual are more prone to heart disease ?

Methods

What methods did you use to analyze the data and why are they appropriate? Be sure to adequately, but briefly, describe your methods.

I did Exploratory data analysis , apart from that I did apply classification technique KNN and decision tree classifiers to predict our target variable

Findings

<Feel free to replicate this slide to show multiple findings>

Present your findings. Include at least one visualization in your presentation (feel free to include more). The visualization should be honest, accessible, and elegant for a general audience.

You need not come to a definitive conclusion, but you need to say how your findings relate back to your research question.

There’s obvious correlation between age and heart disease . The maximum heart rate achieve thalach and presence of disease are also strongly correlated.

Out of the two algorithm the KNN , does well with accuracy of 83.606 , while decision tree classifier has 73.770

Limitations

If applicable, describe limitations to your findings. For example, you might note that these results were true for British Premier league players but may not be applicable to other leagues because of differences in league structures.

Since the size of dataset wasn’t huge hence we should refrain from deducing apply it to large set.

Conclusions

KNN works well on a small data set, as can be seen from the project.

The train-test split does matter a lot to detect accuracy , hence cross validation should definitely done.

Acknowledgements

Where did you get your data? Did you use other informal analysis to inform your work? Did you get feedback on your work by friends or colleagues? Etc. If you had no one give you feedback and you collected the data yourself, say so.

https://archive.ics.uci.edu/ml/datasets/heart+disease

I did not get any feedback from everyone.

References

If applicable, report any references you used in your work. For example, you may have used a research paper from X to help guide your analysis. You should cite that work here. If you did all the work on your own, please state this.

https://github.com/k2datascience/advanced-classification-example/blob/master/HeartDiseaseProject.ipynb

We are going to analyze the data related to heart patients and determine using the KNN classification whether a person is likey to have hear disease or not , also compare two Classification ALgorithm : KNN Classification and Decision Tree Classifiers , how accurate they are in classifying the target vairable i.e. presence of disease. The data is gather from hospitals of Zurich & Basel (Switzerland), Cleveland & LA long beach(USA), Budapest(Hungary).

In [62]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, cross_val_predict,cross
from sklearn.neighbors import KNeighborsClassifier
from sklearn import tree
from sklearn import metrics #Import scikit-learn metrics module for accuracy
import seaborn as sns
```

Reading the data into a data frame.

In [54]:

```
data = pd.read_csv("heart.csv");
```

In [55]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
age         303 non-null int64
sex         303 non-null int64
cp          303 non-null int64
trestbps    303 non-null int64
chol        303 non-null int64
fbs         303 non-null int64
restecg     303 non-null int64
thalach     303 non-null int64
exang       303 non-null int64
oldpeak     303 non-null float64
slope       303 non-null int64
ca          303 non-null int64
thal        303 non-null int64
target      303 non-null int64
dtypes: float64(1), int64(13)
memory usage: 33.2 KB
```

In [56]:
```
data.describe()
```

Out[56]:

	age	sex	cp	trestbps	chol	fbs	restecg	
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202

In [57]:
```
data.head(10)
```

Out[57]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
5	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
6	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
7	44	1	1	120	263	0	1	173	0	0.0	2	0	3	1
8	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
9	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1

```
By looking on the head , we can see which features are categorical and which
are numerical.
Categorical : sex, cp , fbs restecg, exang slope
    sex: sex (1 = male; 0 = female)
    cp: chest pain type
        -- Value 1: typical angina
        -- Value 2: atypical angina
        -- Value 3: non-anginal pain
        -- Value 4: asymptomatic
    trestbps: resting blood pressure (in mm Hg on admission to the
        hospital)
    chol: serum cholestoral in mg/dl
    fbs: (fasting blood sugar > 120 mg/dl)  (1 = true; 0 = false)
    ca: number of major vessels (0-3) colored by flourosopy
    thalach: maximum heart rate achieved
    exang: exercise induced angina (1 = yes; 0 = no)
    oldpeak : ST depression induced by exercise relative to rest
    slope: the slope of the peak exercise ST segment
        -- Value 1: upsloping
        -- Value 2: flat
        -- Value 3: downsloping
    thal: 3 = normal; 6 = fixed defect; 7 = reversable defect
```

In [58]:

```
#Missing Values
data.isnull().sum()
```

Out[58]:

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

Because KNN relies on the majority voting base on class membership of 'k' nearest samples of a given test point. The nearness if based on Euclidean distance, it can place extra emphasis on certain variables that have a larger scale and thus larger differences between point will dominate the outcome of kNN. We would need to do scaling of features.

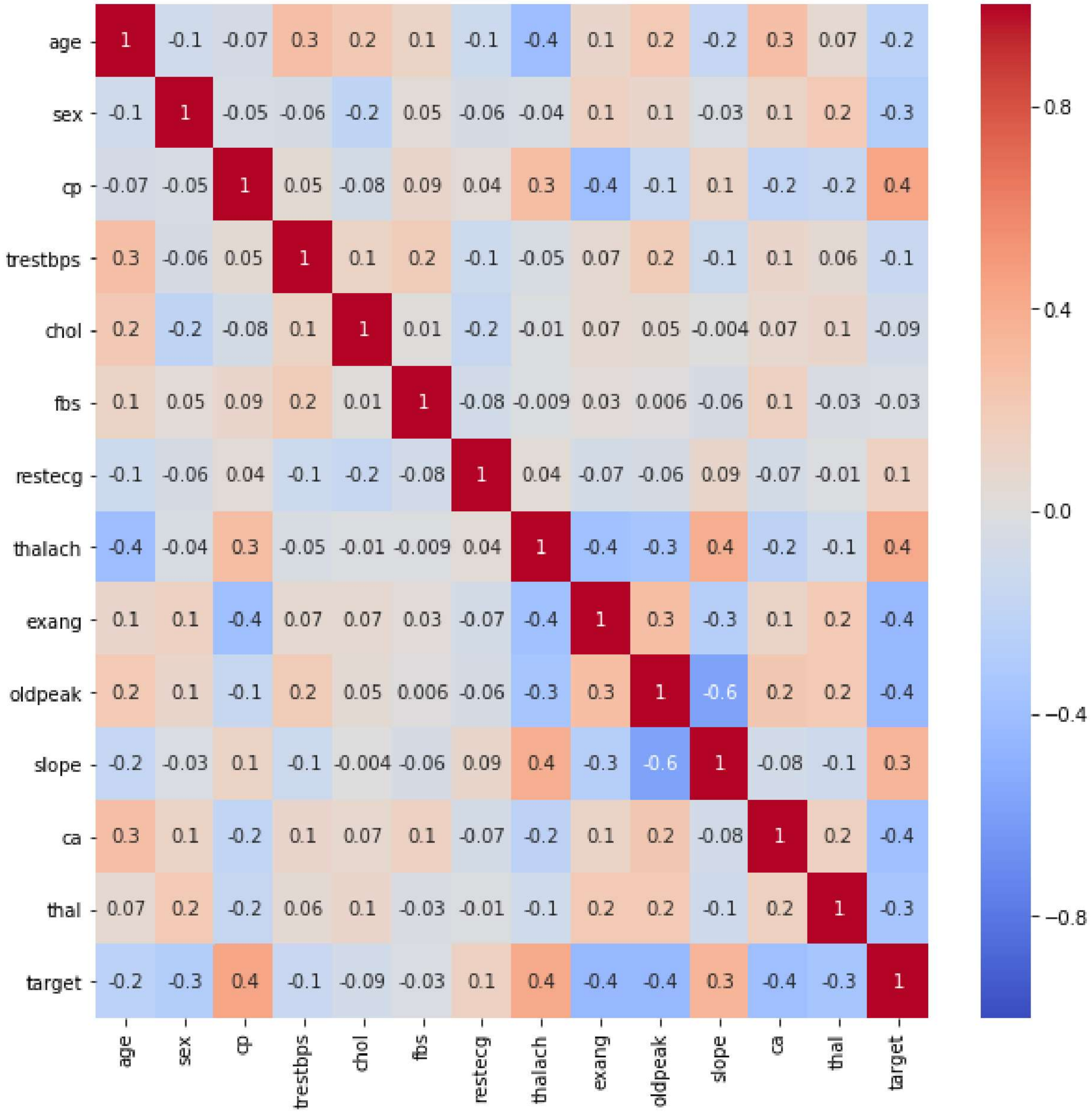
In [59]:

```
cat_col = ['sex', 'cp', 'restecg', 'exang', 'slope', 'ca','thal','fbs','targe
num_col = ['age', 'trestbps', 'chol','thalach','oldpeak']
```

```
To check the correlation between age and the heart disease
```

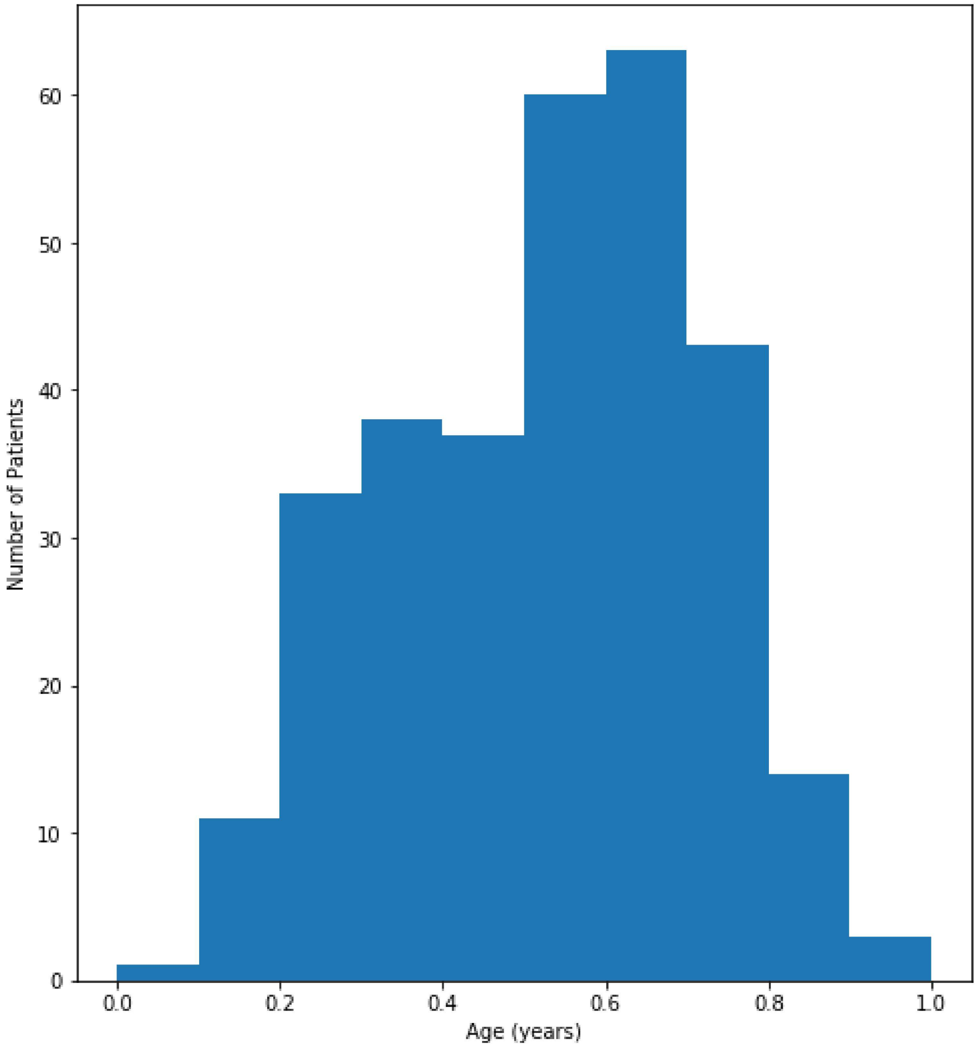
In [106]:
```
fig, ax = plt.subplots(figsize=(10,10))         # Sample figsize in inches
print (sns.heatmap(data.corr(), annot = True,fmt='.1g', vmin=-1, vmax=1, cent
```

```
AxesSubplot(0.125,0.125;0.62x0.755)
```



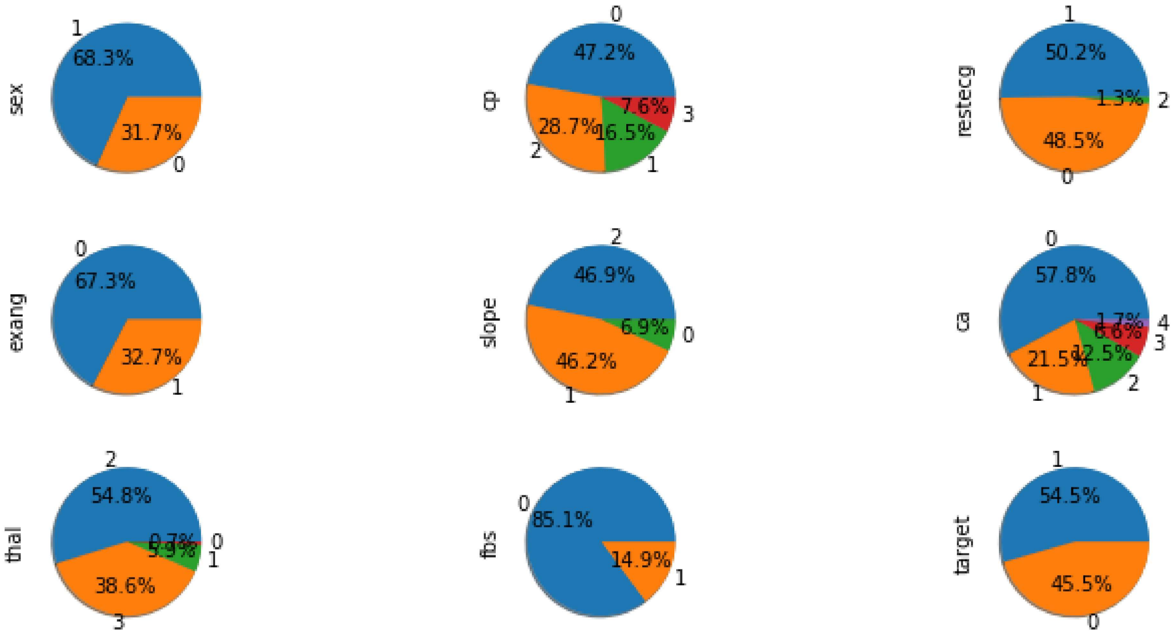
```
In [119]: fig, axes = plt.subplots( figsize=(8,8) )
          plt.subplots_adjust( wspace=0.20, hspace=0.20, top=0.97 )
          plt.hist(data.age)
          plt.xlabel("Age (years)")
          plt.ylabel("Number of Patients")
```

```
Out[119]: Text(0, 0.5, 'Number of Patients')
```



In [60]:

```
plt.figure(figsize=(12,18))
count = 1
for cols in cat_col:
    plt.subplot(9, 3, count)
    data[cols].value_counts().plot.pie(shadow=True,autopct='%1.1f%%')
    count +=1
```



Scaling the other Numerical Variables.

In [64]:

```
scaler = MinMaxScaler()
data[num_col] = scaler.fit_transform(data[num_col])
```

Now that we have scaled the data, we can start the clustering process. For the kNN algorithm.Now that we have to separate the features and the target variable , befor applying the kNN.

Type *Markdown* and LaTeX: α^2

In [65]:

```
train = data.drop(["target"],axis=1)
train_ = data["target"]

X_train = train.values
y_train = train_.values
```

In [66]:

```
train_x, test_x,train_y,test_y = train_test_split(X_train,y_train,test_size
print("Train dataset shape: {0}, \nTest dataset shape: {1}".format(train_x.sh
```

```
Train dataset shape: (242, 13),
Test dataset shape: (61, 13)
```

In [96]:

```
test_scores = []
train_scores = []
Misclassified_sample = []
for i in range(1,15):

    knn = KNeighborsClassifier(i)
    knn.fit(train_x,train_y)
    y_pred = knn.predict(test_x)
    train_scores.append(knn.score(train_x,train_y))
    test_scores.append(knn.score(test_x,test_y))
    Misclassified_sample.append((test_y != y_pred).sum())
print("Misclassified_sample = ", Misclassified_sample)
```

```
Misclassified_sample =  [10, 13, 7, 11, 9, 7, 8, 8, 10, 10, 9, 8, 10, 10]
```

In [97]:

```
# Lowest number of samples for K=3

KNN_classifier = KNeighborsClassifier(n_neighbors=3)

# Fitting the values fo X and Y
KNN_classifier.fit(train_x, train_y)

#Predicting the test values with Model
prediction =  KNN_classifier.predict(test_x)
```

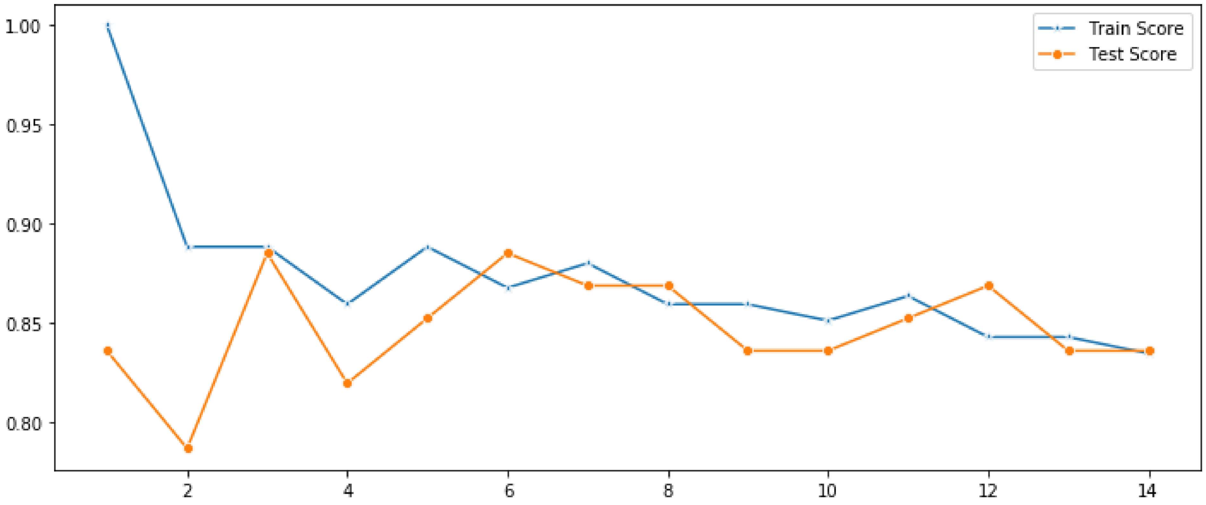
In [98]:

```


#Score/Accuracy
print("Accuracy --> ", knn.score(test_x,test_y)*100)

## score that comes from testing on the datapoints that were split in the beg

max_test_score = max(test_scores)
test_scores_ind = [i for i, v in enumerate(test_scores) if v == max_test_scor
print('Max test score {} % and k = {}'.format(max_test_score*100,list(map(lam
```

```
Accuracy -->  83.60655737704919
Max test score 88.52459016393442 % and k = [3, 6]
```

In [89]:

```
plt.figure(figsize=(12,5))
p = sns.lineplot(range(1,15),train_scores,marker='*',label='Train Score')
p = sns.lineplot(range(1,15),test_scores,marker='o',label='Test Score')
```

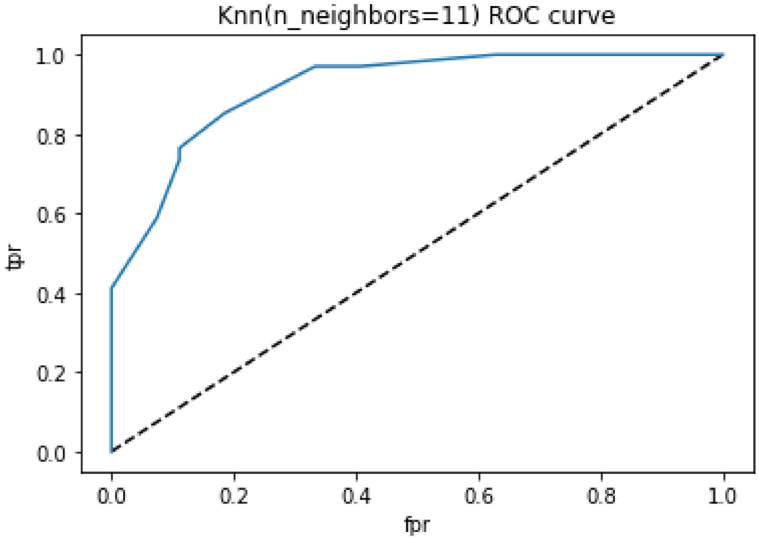


In [90]:

```
from sklearn.metrics import roc_curve
y_pred_proba = knn.predict_proba(test_x)[:,1]
fpr, tpr, thresholds = roc_curve(test_y, y_pred_proba)
```

In [91]:

```
plt.plot([0,1],[0,1],'k--')
plt.plot(fpr,tpr, label='Knn')
plt.xlabel('fpr')
plt.ylabel('tpr')
plt.title('Knn(n_neighbors=11) ROC curve')
plt.show()
```


Knn(n_neighbors=11) ROC curve

In [99]:

```
#Decision Tree Clasifier
t = tree.DecisionTreeClassifier()
t.fit(train_x,train_y)
y_pred = t.predict(test_x)
#Score/Accuracy
print("Accuracy --> ", t.score(test_x,test_y)*100)
```

```
Accuracy -->  73.77049180327869
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: